ISS Projekt 2019 / 20

Honza Černocký, ÚPGM FIT VUT

November 18, 2019

1 Úvod

Cílem projektu je udělat jednoduchý systém pro vyhledávání v audiu pomocí akustického vzoru, Query by Example (QbE). Na začátku projektu nahrajete 10 vět podle známé americké řečové databáze TIMIT[1]. Pak si nahrajete dvě klíčová slova a ta budete v nahraných 10ti větách hledat.

Soubor vět pro nahrání je pro každého individuální a je v:
`https://www.fit.vutbr.cz/study/courses/ISS/private/proj2019-20/prompts/xlogin00.txt`, kde `xlogin00` je Váš login.

Projekt je možno řešit v Pythonu, Matlabu, Octave, jazyce C nebo v libovolném jiném programovacím nebo skriptovacím jazyce. Je možné použít libovolné knihovny. Projekt se nezaměřuje na "krásu programování", není tedy nutné mít vše úhledně zabalené do okomentovaných funkcí, ošetřené všechny chybové stavy, atd. Důležitý je výsledek. Kód **musí být možné spustit na školním Linuxu nebo Windows a musí prokazatelně produkovat výsledky obsažené ve Vašem protokolu.**

2 Odevzdání projektu

bude probíhat do informačního systému WIS ve dvou souborech:

1. `xlogin00.pdf` (kde "xlogin00" je Váš login) je protokol s řešením.
   - V záhlaví prosím uveďte své jméno, příjmení a login.
   - Pak budou následovat odpovědi na jednotlivé otázky — obrázky, numerické hodnoty, komentáře.
   - U každé otázky uveďte stručný postup - může se jednat o kousek okomentovaného kódu, komentovanou rovnici nebo text. Není nutné kopírovat do protokolu celý zdrojový kód. Není nutné opisovat zadání či teorii, soustřeďte se přímo na řešení.
   - Pokud využijete zdroje mimo standardních materiálů (přednášky, cvičení a studijní etapa projektu ISS), prosím uveďte, odkud jste čerpali.
   - Protokol je možné psát v libovolném systému (Latex, MS-Word, Libre Office, ...), můžete jej psát i čitelně rukou, dolepit do něj obrázky a pak oskenovat. Protokol může být česky, slovensky nebo anglicky.
   - Doporučená délka protokolu jsou 2 strany + obrázky, případně 3, pokud se rozhodnete řešit bonusový úkol.
2. `xlogin.tar.gz` je komprimovaný archiv obsahující následující adresáře:
   - `/src` - Vaše zdrojové kódy – může se jednat o jeden soubor (např. `moje_reseni.m`), o více souborů či skriptů nebo o celou adresářovou strukturu.
   - `/sentences` - nahrané věty se správnými názvy souborů, ve formátu WAV, na vzorkovací frekvenci 16 kHz, bitová šířka 16 bitů, bez komprese. V protokolu Vás prosíme o informaci, k čemu můžeme data použít, viz níže.
   - `/queries` - soubory `q1.wav` a `q2.wav` s hledanými klíčovými slovy, ve stejném formátu jako výše, zbavené ticha.
   - `/hits` - soubory s nalezenými nejpravděpodobnějšími výskyty klíčových slov, ve stejném formátu jako výše, "vykousnuté" z příslušné věty.
3. Projekt je **samostatná práce**, proto budou Vaše zdrojové kódy křížově korelovány a v případě silné podobnosti budou vyvozeny příslušné závěry.
4. Silná korelace s kódy ze studijní etapy projektu je v pořádku, nemusíte tedy měnit názvy proměnných, připisovat zbytečné komentáře, atd.

[1] `https://catalog.ldc.upenn.edu/LDC93S1`

3 Zadání

1. **[1 bod]** Namluvte věty podle Vašeho osobního souboru, na řádku je vždy jméno souboru a to, co máte říci, např.:

   ```
   sa1.wav: She had your dark suit in greasy wash water all year.
   sa2.wav: Don't ask me to carry an oily rag like that.
   si1972.wav: Perfect, he thought.
   ...
   ```

   Nahrávat můžete na čemkoliv, stačí běžný smartphone nebo notebook, není nutné hledat studiový mikrofon a HiFi zvukovou kartu. Doporučujeme smartphony, mívají lepší mikrofon než notebooky. Volte běžné klidné prostředí (byt, kancelář), není nutné shánět odhlučněnou místnost. Hovořte svou běžnou angličtinou, není nutné snažit se o anglický nebo americký akcent. Požadovaný formát pro data je:

   - standardní WAV bez komprese,
   - vzorkovací frekvence F_s =16kHz,
   - mono (1 kanál),
   - 16 bitů na 1 vzorek.

   Pokud Váš nahrávací software takový formát nepodporuje, můžete nahrávat na vyšší vzorkovací frekvenci a pak převést, např. pomocí běžného programu `ffmpeg`, pro běžné audio-nahrávky z Androidu např. takto:
   `ffmpeg -i Sa1.m4a  -ar 16000 -ac 1 -acodec pcm_s16le sa1.wav`

   Počáteční písmena názvů souborů nechť jsou prosím malými písmeny. Výsledné nahrávky zkontrolujte poslechem, a pokud v některé objevíte vážný problém (přeřek, bouchnutí, atd.), prostě ji opakujte.

   Do protokolu uveďte tabulku s názvy souborů a délkou vět ve vzorcích a v sekundách. Délku získáte např. pomocí programu `soxi` takto: `soxi sa1.wav`

   Takto nahraná řečová data jsou cenným materiálem, do protokolu prosím uveďte, jak je můžeme použít (preferujeme samozřejmě (b) nebo (c), ale nemůžeme Vás nutit a body za to nejsou...)

   (a) pouze pro vyhodnocení tohoto ISS projektu.

   (b) pro (a) a pro výzkum a vývoj v rámci řečové skupiny na FITu BUT Speech@FIT.

   (c) pro (a), (b) a pro tvorbu volně dostupné databáze "Czenglish TIMIT". Ta nebude v žádném případě obsahovat žádná Vaše osobní data (jméno, příjmení ani login).

   Kontrola: v adresáři `/sentences` je 10 WAV souborů se správnými jmény, F_s, bitovou šířkou, které obsahují to, co mají. Protokol obsahuje tabulku a informaci o použití dat.

---

2. **[1 bod]** Z textu k nahrávání vyberte dvě klíčová slova, která budou hrát roli "queries". Je dobré, aby měla alespoň 3 slabiky, čím delší slovo, tím lépe se detekuje. Tato slova nahrejte a uložte ve stejném formátu, jako je popsáno výše, do `q1.wav` a `q2.wav`. Před uložením **zbavte slova okolního ticha** – s tichem by se Vám velmi špatně detekovala. To můžete provést v prostředí, kde pracujete (Octave, Python, ...), ale můžete klidně použít libovolný software pro editaci audia – Audacity, GoldWave, WaveSurfer[2] nebo jiný. Opět zkontrolujte poslechem.

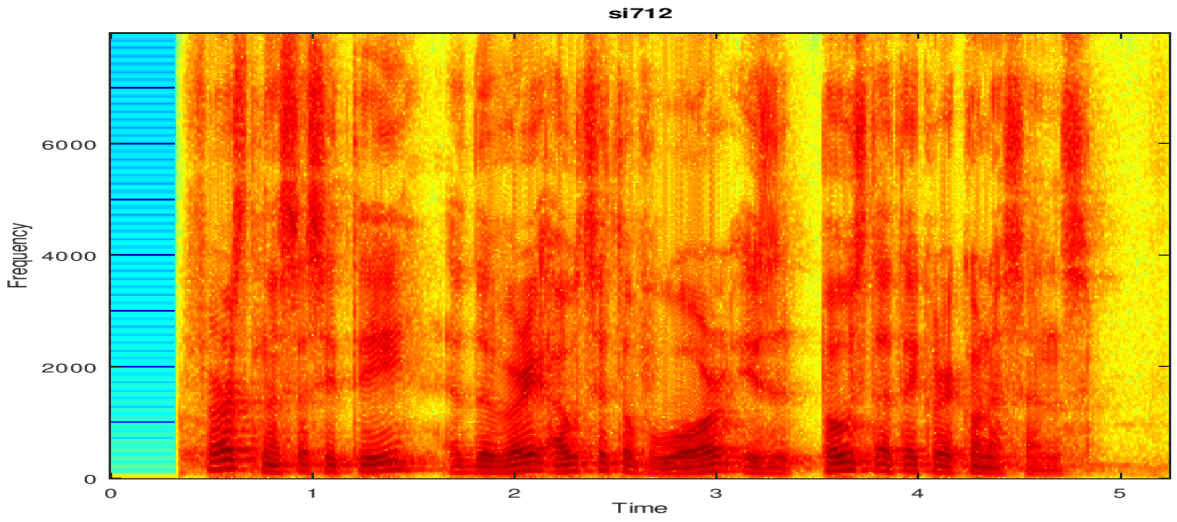
   Do protokolu uveďte tabulku s vybranými slovy pro `q1.wav` a `q2.wav` a s jejich délkou ve vzorcích a v sekundách.

   Kontrola: v adresáři `/queries` jsou `q1.wav` a `q2.wav` se správnými jmény, F_s a bitovou šířkou. V protokolu je tabulka. Pokud jsou queries delší než věty, něco je špatně...

---

3. **[1 bod]** Naučte se produkovat a zobrazovat spektrogram. Jedná se o 2D obrázek, kde vodorovně je čas, svisle frekvence a barva značí hodnotu spektra na dané frekvenci v daném čase, viz příklad.

[2] Můj oblíbený, není nutné instalovat a rozjede se prakticky všude `https://sourceforge.net/projects/wavesurfer/`



Technicky výpočet probíhá následovně

(a) před dalším zpracováním je dobré signál ustřednit pomocí odečtení střední hodnoty. Pokud v něm byla (škodlivá) stejnosměrná složka, spolehlivě ji tak zničíte. V Matlab/Octave: `s = s - mean(s)`

(b) signál rozdělíme na segmenty (rámce) o délce 25 ms s překrytím 15 ms, takže posun mezi jednotlivými rámci je 10 ms. Dostaneme jich tedy 100 za sekundu.

(c) vybereme jeden rámec pomocí okénkové funkce, nejčastěji se používá tzv. Hammingovo okno, které utlumí signál na okrajích.

(d) vybraný rámec se doplní nulami tak, aby se pak dobře počítala rychlá Fourierova transformace, tedy na mocninu dvou. Takto doplněný rámec označme $x[n]$ a jeho délku N.

(e) vypočte se rychlá Fourierova transformace, která dá N komplexních koeficientů $X[k]$. Ty omezíme jen na $\frac{N}{2}$, protože nemá cenu nic dělat nad polovinou vzorkovací frekvence.

(f) z výsledku se vypočte logaritmické výkonové spektrum[3]

$$P[k] = 10 \log_{10} |X[k]|^2$$

(g) výsledek se uloží jako sloupec do matice se spektrogramem a vhodně (černobíle nebo barevně) zobrazí.

Lze použít funkce dostupné v různých knihovnách, např. můj kód (Octave) vypadá takto:

```
Fs = 16000; N = 512; wlen = 25e-3 * Fs; wshift = 10e-3*Fs; woverlap = wlen - wshift;
win = hamming(wlen); %plot(win);
f = (0:(N/2-1)) / N * Fs;
t = (0:(1 + floor((length(x) - wlen) / wshift) - 1))* wshift/Fs;  % minus one as of Matlab ...
X = specgram (x, N, Fs, win, woverlap);
imagesc(t,f,10*log(abs(X).^2));
set (gca (), "ydir", "normal"); xlabel ("Time"); ylabel ("Frequency"); colormap(jet);
```

Klidně jej použijte nebo použijte kód v Pythonu od Katky Žmolíkové[4], ale měli byste přesně vědět, co spektrogram obsahuje.

Do protokolu vložte spektrogram libovolné věty, titulek obrázku ať je název věty. Osy nechť jsou skutečně v sekundách a Hertzích.

Kontrola: v protokolu je obrázek spektrogramu s pěknými osami a názvem. Máte nachystanou funkci/kód pro výpočet spektrogramu.

---

[3] Prosím opravdu nezapomeňte na absolutní hodnotu, u komplexních čísel c^2 **není totéž** co $|c|^2$!

[4] `http://www.stud.fit.vutbr.cz/~izmolikova/ISS/project/`

4. **[1 bod]** Napište funkci pro výpočet parametrů (features). Těmi bude nutné popsat query a prohledávanou větu. Pro každý rámec je nutné vyprodukovat jeden vektor parametrů. Parametry můžete vymyslet jaké chcete, my doporučujeme výstupy lineární banky filtrů, která produkuje pro každý rámec $N_c = 16$ koeficientů[5] jednoduše tak, že posčítá $B = \frac{256}{N_c}$ koeficientů logaritmického výkonového spektra:

$$f_0 = \sum_{k=0}^{B-1} P[k], \qquad f_1 = \sum_{k=B}^{2B-1} P[k], \quad \ldots \quad f_{B-1} = \sum_{k=256-B}^{256-1} P[k]$$

Výsledné features je potřeba uložit do matice, jeden sloupec pro každý rámec.

Výpočet se dá popsat pěkně maticově, je-li matice se spektrogramem $\mathbf{P}$ a chceme-li matici parametrů $\mathbf{F}$, jde výpočet realizovat maticovým násobením:

$$\mathbf{F} = \mathbf{AP}.$$

Vymyslete, jak naplnit matici $\mathbf{A}$.

Do protokolu popište Vámi zvolený výpočet parametrů, v případě, že použijete popsanou lineární banku filtrů, popište, jak naplnit matici $\mathbf{A}$.

Kontrola: v protokolu je popis výpočtu parametrů, případně tvorba matice $\mathbf{A}$. Máte nachystanou funkci/kód pro výpočet parametrů.

---

5. **[1 bod]** Napište funkci pro výpočet skóre klíčového slova (query) pro jednu větu. Její hodnota by měla udávat pravděpodobnost začátku nebo konce klíčového slova v závislosti na čase. Výpočet skóre můžete vymyslet jaký chcete, my doporučujeme nechat matici parametrů query $\mathbf{Q}$ "projíždět" kolem matice paramterů věty $\mathbf{F}$ a pro každou poteciální polohu query *pp* počítat jednu hodnotu skóre. Jako vzdálenost docela funguje součet Pearsonových korelačních koeficientů mezi jednotlivými vektory:

$$d_{pp} = d(\mathbf{Q}, \mathbf{F}_{(pp:pp+N_q-1)}) = \sum_{k=0}^{N_q-1} p(\mathbf{q}_k, \mathbf{f}_{pp+k}),$$

kde N_q je počet vektorů v query (resp. její délka v setinách sekundy), N_c je počet koeficientů, $\mathbf{q}_k$ a $\mathbf{f}_k$ jsou sloupce matice parametrů query a věty a notace $\mathbf{F}_{(i:j)}$ znamená "sloupce i až j z matice $\mathbf{F}$". Pearsonův korelační koeficient je definován jako

$$p(\mathbf{a}, \mathbf{b}) = \frac{\sum_{i=0}^{N_c-1}(a_i - \bar{a})(b_i - \bar{b})}{\sqrt{\sum_{i=0}^{N_c-1}(a_i - \bar{a})^2}\sqrt{\sum_{i=0}^{N_c-1}(b_i - \bar{b})^2}}.$$

$\bar{a}$ je průměr všech prvků vektoru $\mathbf{a}$, podobně pro $\mathbf{b}$. Matematicky hloubaví dokáží jistě vzorec interpretovat jako "skalární součin ustředněného vektoru $\mathbf{a}$ s ustředněným vektorem $\mathbf{b}$ lomeno geometrickým průměrem norem ustředněného vektoru $\mathbf{a}$ a ustředněného vektoru $\mathbf{b}$;)

Potenciální poloha query *pp* může postupovat po jednom vektoru, ale je možné, že výpočet by pak byl hodně pomalý, zkoušeli jsme (docela úspěšně) zvyšování *pp* po 5 rámcích (tedy po 0.05 s).

Do protokolu popište Vámi zvolený výpočet skóre. Můžete například vložit komentovaný kód.

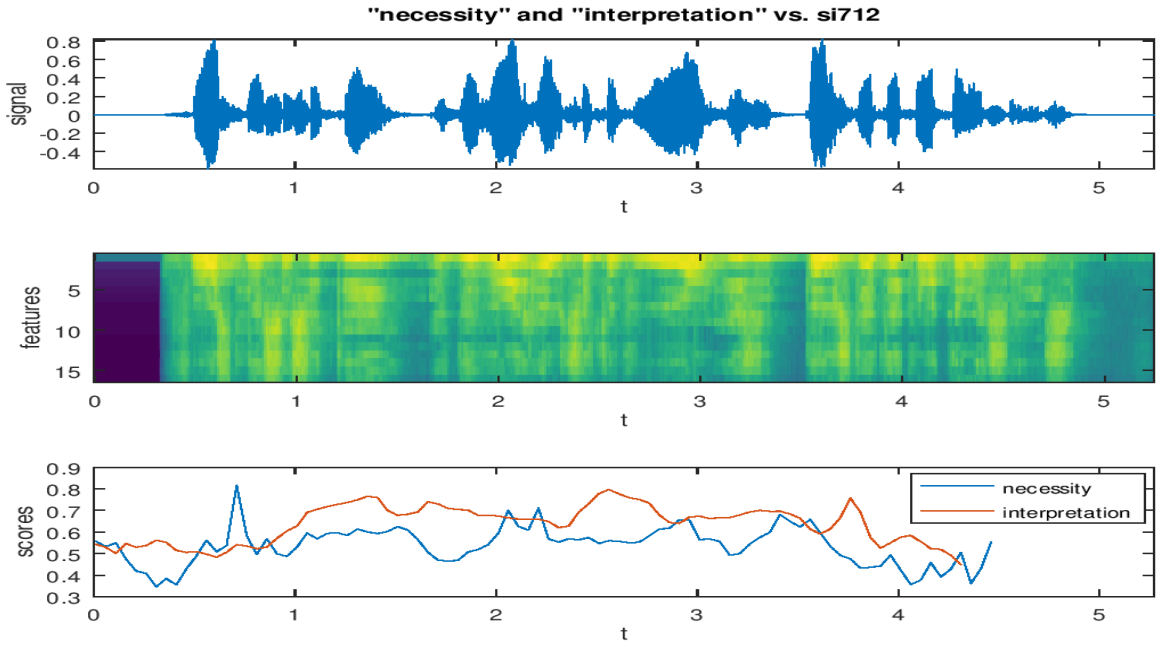
Kontrola: v protokolu je popis výpočtu skóre. Máte nachystanou funkci/kód pro výpočet průběhu skóre query v zadané větě.

---

6. **[3 body]** Toto je hlavní grafický výstup projektu. Přiložte deset obrázků, každý bude mít v titulku název věty a bude sestávat ze tří částí, viz příklad:

   - signál,
   - matice parametrů věty,
   - průběh skóre klíčových slov `q1.wav` a `q2.wav` v průběhu věty. Označte, který průběh je pro které slovo.

[5] Proč tak málo a ne původních 256 ? Plné spektrum popisuje řeč hodně detailně, takže by nemuselo (ani pro stejné hlásky) dojít ke shodě, menší počet parametrů je "bezpečnější".



Časové osy nechť jsou pro všechny obrázky v sekundách. To Vás může trochu potrápit, protože u signálu, parametrů a skóre má časová osa jiný krok (u signálu jeden vzorek, u parametrů jeden rámec a u skóre jeden nebo několik rámců). Hodnot skóre navíc může být méně než rámců věty, obvykle to asi bude $N_v - N_q$, protože query "nevysunujeme" mimo větu. Snažte se, aby časové osy navzájem "seděly", viz příklad.

Kontrola: v protokolu je 10 obrázků se názvy, signály, parametry a průběhy skóre, je označené, co je co, všechny mají pěknou časovou osu.

---

7. [**1 bod**] Popište, jak budete z průběhu skóre určovat, zda query na daném místě je nebo není. Zřejmě bude nutné použít nějaký rozhodovací práh, popište, jak ho určíte. Nemusí se jednat o velkou vědu, stačí analyzovat to, co vidíte. Napište jeho hodnotu(y) pro `q1.wav` a `q2.wav`.

   Kontrola: v protokolu je popis určení prahu(ů) a jejich hodnoty pro obě dvě queries.

---

8. [**2 body**] Uveďte výsledky - tabulku s jednotlivými větami a klíčovými slovy, pro každou kombinaci výskyt ano/ne a pokud ano, od kterého do kterého vzorku se tam query nachází. Nálezy "hits" extrahujte do krátkých WAV souborů a uložte do adresáře `/hits` s nějakými inteligentními jmény, např. `q1_sa1.wav`, `q2_si1972.wav`. Poslechněte si, zda jste se "trefil/a".

   Kontrola: v protokolu je tabulka s nálezy a v adresáři `/hits` jsou WAV soubory s nálezy queries.

---

9. [**1 bod**] Závěr: zpracování řeči není jednoduché a je možné, že Vaše výsledky nebudou "nic moc" ani když budete mít vše dobře naprogramované. Krátce zhodnoťte zda Váš detektor funguje nebo ne, kde funguje, kde selhává a jak by se dal zlepšit.

   Kontrola: v protokolu je závěr, přečetli jste si ho po sobě.

4 Bonusový úkol

je nepovinný, není hodnocen body, ale nejzajímavější řešení vyhraje láhev dobrého francouzského červeného vína.

Je zřejmé, že největším problémem navrhovaného postupu je **rozdílné časování** - v query stačí říci jednu samohlásku o trochu delší, časování se rozjede, vektory, které by měly být stejné, nepadnou na sebe a hodnota skóre jde do háje. **Zkuste s časováním něco udělat, aby se výsledky zlepšily**. Zde je několik návrhů, ale můžete přijít s čímkoliv dalším !

1. Pro každou query vyslovte několik verzí — normálně, pomalu, rychle, s různými délkami samohlásek.

2. Nahrajte query jen jednou, ale použijte umělé roztáhnutí nebo smrsknutí - to můžete naimplementovat sami na úrovni matice parametrů **Q** nebo pomocí software pro úpravu audia. Např. populární efekt `atempo=...` v `ffmpeg` nebo podobný efekt v `sox`.

3. použijte pro časové zarovnání matice parametrů query a vybrané matice parametrů z věty dynamické borcení času - Dynamic Time Warping, DTW[6]

4. Pro další nápady ohledně QbE se můžete podívat do článku "Comparison of methods for language-dependent and language-independent query-by-example spoken term detection" Javiho Tejedora a Igora Szökeho[7], případně zajít za mnou, za Igorem nebo za kterýmkoliv dalším členem skupiny BUT Speech@FIT.

Další zlepšení se netýká časování, ale normalizace parametrů: zkuste průběhy výstupů každého filtru **v čase**[8] normalizovat tak, aby měly nulovou střední hodnotu a jednotkovou varianci. Říkáme tomu MVN (Mean and Variance Normalization). Pro query to asi uděláte jen jednou. U prohledávané věty je možné normalizovat jednou celou větu nebo normalizaci provést pro každou srovnávanou pod-matici $\mathbf{F}_{(pp:pp+N_q-1)}$. Při výpočtu parametrů pro normalizaci je dobré dát pozor na velká množství ticha, které může střední hodnoty i variance slušně zkreslit — je dobré udělat si jednoduchý detektor ticha (založený na energii rámce nebo na součtu hodnot spektrogramu pro jeden rámec) a ticho vyloučit z odhadu střední hodnoty a variance.

Pokud se rodhodnete pro řešení bonusového úkolu, v protokolu své řešení krátce popište a uveďte jeden (ne deset) obrázek, kde srovnáte skóre základního detektoru a Vaší vylepšené verze (Vašich vylepšených verzí).

Na závěr bonusového úkolu můžete zkusit vyhledávat ne své klíčové slovo, ale query vypreparovanou **ze skutečné databáze TIMIT** v souboru
`https://www.fit.vutbr.cz/study/courses/ISS/private/proj2019-20/bonus_timit_query/xlogin00.wav`
Jednoduchý postup popsaný v zadání na kombinaci "query ze skutečného TIMITu" vs. "věta z Czechglish TIMITu" kvůli rozdílům v řečníkovi, akcentu, nahrávacím zařízení a prostředí totálně selhává. Uvidíme, jak si poradí Vaše vylepšená verze !

[6] viz studijní opora kursu ZRE `https://www.fit.vutbr.cz/study/courses/ZRE/public/opora/zre_opora.pdf` a laboratorní úloha `https://www.fit.vutbr.cz/study/courses/ZRE/public/labs/05_dtw_hmm/` (jen část DTW).

[7] `https://www.fit.vut.cz/research/publication/10179/`

[8] Ne přes všechny koeficienty filtru ! To už dělá Pearsonův koeficient.